Design and Implementation of Smartcard-based Secure E-Mail Communication

Hsien-Hau Chen; Yung-Sheng Chen
Division of Research & Development
NexSmart Technology, Inc.
3F-2, No.23, Sec.6, Min Chuan E. Road.,
Taipei, Taiwan 114, ROC

Hsia-Ling Chiang[†], Ph.D.
Department of Electrical Engineering
Kuang Wu Institute of Technology
No. 151, I-te Street, Peitou,
Taipei, Taiwan 112, ROC

Chung-Huang Yang[†], Ph.D.
Inst. of Information & Computer Education
National Kaohsiung Normal University
No. 116, Ho Ping 1st Road,
Kaohsiung, Taiwan 802, R.O.C.

Abstract

E-mail system is by far the most widely used application in the Internet. However, mainly due to the lack of communication security, sensitive messages could not transmit securely over open networks using off-the-shell e-mail systems. In this paper, a new secure e-mail system is proposed and implemented to extend the popular Microsoft Outlook e-mail software with flexible security services and to combine these services tightly with smartcards. The enhanced security services include data confidentiality, authentication of message originator and recipient, data integrity, and non-repudiation.

The proposed system provides two approaches for secure e-mail communication, one is base on the certification authority (CA) and the other is base on the keys distribution center (KDC), such that a complete solution may be satisfied for both open public and private enterprise. Windows-Based smart cards, NexCard 2.0, is adopted as portable security tokens to store private key for generating digital signature, to store multiple digital certificates issued from the CAs and to store the master key shared with the KDC. We also designed and implemented cryptographic libraries, CSP 2.0 and PKCS#11, which is need for secure interaction of smartcard module with applications.

Keywords: smartcard, e-mail, digital certificate, AES, key management, encryption, authentication, digital signature

I.INTRODUCTION

Electronic mail (e-mail) has become the most widely used tool for communication, in today's popular Internet environment. It is naturally the target of many attacks, and therefore the security of e-mail is emphasized [1-2]. In order to handle the problem, many security schemes have developed, such as the Pretty Good Privacy (PGP) [3] scheme or the Privacy Enhanced Mail (PEM) [4] or the S/MIME [5]. These secure e-mail schemes provide the cryptographic security services for electronic messaging applications, including authentication, message integrity and non-repudiation of origin and privacy and data security. There are other schemes that, for example, establish a certified e-mail protocol to protect both the sender and the receiver [6]. However, two problems make existing secure e-mail systems not to apply popularly. The one is that the systems are not thorough security protection and have still doubts. The other is that the applied area cannot cover in many ways so the usage is very inconvenient.

Regarding to security protection, secure e-mail makes mainly characteristic of asymmetric key to reach the requirements of the cipher and communication security. Nevertheless, the cryptographic key is still stored in the computer hard disk or external store media that are not secure enough for abilities of the protection. The hackers and viruses from the network might apply the possible leaks in store media to make an intrusion. Intrusion system and human errors could reduce substantially by improving protection in store media and the whole safety is increased. Therefore, the store medium that usage is off-line, convenient and there is high protection mechanism again indeed is to construct the necessary consideration of the high safety secure e-mail. Moreover, the entity invaded and human errors in actual environment reduce much the integrated security.

Another issue is regarding to make use of convenience. The enterprise needs three kinds of communication, to the internal security communication in enterprise, to the external security communication outside, to the

[†] This research was supported by the NexSmart Technology, Inc., Taipei, Taiwan, R.O.C.

normal communication of the inner or outer enterprise. The familiar secure e-mail systems in enterprise transmit the mails of encryption and signing through the exclusive CA of self-establishment or trusted third party CA to apply for certificates directly. Usually the CA is only a part of much more complex and hierarchical structure known as Public Key Infrastructure (PKI). Among the more relevant components of PKI we distinguish CA, Registration Authority (RA), repositories of E-terms that is of Certificate Practice Statements (CPS) and Certificate Policies, repositories of certificates and Certificate Revocation List (CRL). The establishment and maintenance of exclusive CA have high cost and compare for inconvenience to outward communication. However, the certificates of third party CA acquire hard to trust well and valid management in interior enterprise. The mechanisms of CAs trust each other make performance to decrease. The KDC does not need to certified, so speed of process is better. Constructing a convenience and comprehensive systems will play important factors of the secure e-mail systems.

In this paper, we propose the client of Smartcard-based Secure E-mail System that integrated into Microsoft (MS) Outlook. In addition to establishment of Keys Distribution Center (KDC) system in inner part of the enterprise, the system merges to use the external trusted third-party CA certificate in order to reach to break the space obstacle. Furthermore, the smart card stores the certificates, private key and master key of KDC generation. The dual need that to reach the high security and convenience. The system architecture is shown in figure 1.

II. SYSTEM OPERATIONS

The main conception of Smartcard-based Secure E-mail System is to combine the public-key certificate of the CA and the master key of the KDC that deposited in high secure and portable smart card. When users need to send the messages of encryption or signing, they must obtain the cipher key to proceed to encrypt through the smart card first. In addition, the message digest must created by hashing message in the computer, then deliver after computing the message digest with the SHA1 hash algorithm in the smart card. So as not to the private key in the computer grabbed, increase the security degree of the system. For attaining these functions, our system keeps the original functions of mail server, and develops the add-in library in mail clients. With the enhanced MS Outlook functions, one could access the certificate and the cipher key in the smart card through Cryptographic Service Provider (CSP) [7] in the operation system, and complete the mail of encryption and signing.

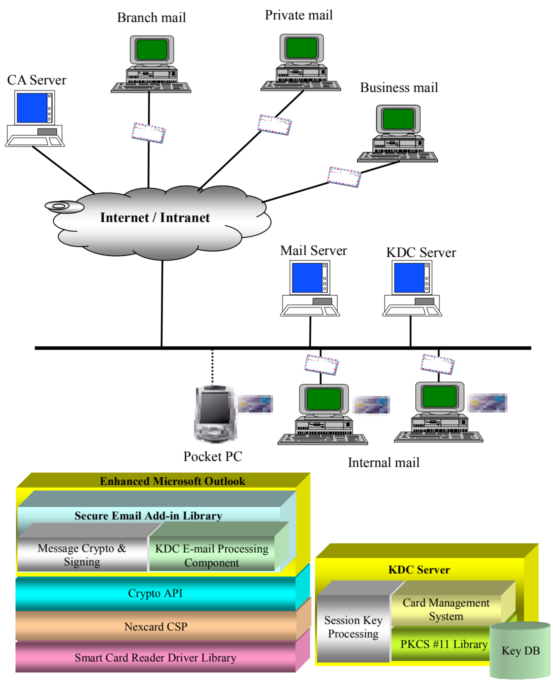


Figure1. System Architecture Diagram

Moreover, we use the KDC e-mail processing component add-in Outlook to deal with sending and receiving secure messages. As regards KDC server, the main functions are management of employees' master keys, such as issuing, revoking the smart card, and the period of validity. In addition, the keys store in the HSM or through encryption in the database for keeping employees from leaving office or losing the keys. The KDC another main work each to encrypt the e-mail transmission generates a random session key and time stamps immediately, and merges to confirm the data of the sender's identify that must decrypt by the master key by me.

Secure e-mail is one of the most exciting public-key-enabled applications because it allows users to share information confidentially and to trust that the integrity of the information maintained during transit. By using MS Outlook, a user can select a public-key certificate issued by a trusted certificate authority to use for digitally signing and decrypting secure messages. By publishing the user's certificate to a public directory in the enterprise or on the Internet, other users in a company or on the Internet can send encrypted e-mail to the user, and visa-versa. For the e-mail of the external enterprise, still use to the certificate of CA that we deposit the private key and master key of KDC together in the smart card.

2-1. SEND E-MAIL PROCEESS

The options including Normal, CA and KDC methods provide for using to send the messages. Using message encryption and signing by CA method, the system will request to insert into smart card of the sender. The part of message signing, the messages contents used to compute a hash value that is message digest. The digital signature is generated by sender's private key in smart card with message digest as a "digital fingerprint" that verifies the message has not been tampered in communication. The part of message encryption, the system generates one-time symmetric session key (K_S) and encrypts message contents in K_S, and then encrypt K_S in the receiver’s public key. Complete the message encrypting and signing, and then send the e-mail to receiver. It is the operation as follow:

$$P_{pubB}[K_S] \parallel En[K_S , [M \parallel P_{priA}[H(M)]]]$$

The sender selects KDC method option to encrypt the message. Before send the message, the system merges sender identity and receiver identity to request KDC for generating a one-time random session key. The KDC receives and analyzes this request, then generates the random session key (K_S) and time stamp (T_S). The K_S and T_S are encrypted by sender's master key in smart card. The KDC also encrypts the receiver's identity and K_S by recipient's master key. Then it replies to the sender above the two items.

$$En[K_A , (T_S\|K_S)] \parallel En[K_B , (K_S \parallel ID_A)]$$

The sender receives from the replied data of the KDC by K_A, and obtains the decrypted random session key Ks. Also confirm replied time whether within the allowed scope . It encrypts the message by the K_S and appends to receiver’s identify to deliver the message to receipt.

$$En[K_S , M] \parallel En[K_B , (K_S \parallel ID_A)]$$

The sending e-mail process diagram is illustrated in figure 2.

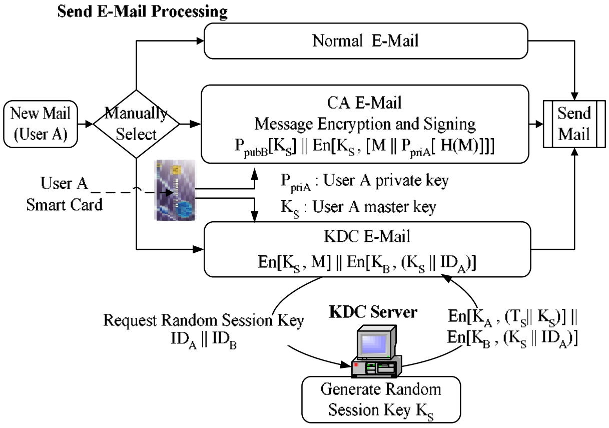

Send E-Mail Processing
Normal E-Mail
New Mail (User A)
Manually Select
CA E-Mail
Message Encryption and Signing
$P_{pubB}[K_S] \parallel En[K_S , [M \parallel P_{priA}[H(M)]]]$
Send Mail
User A Smart Card
P_{priA} : User A private key
K_S : User A master key
KDC E-Mail
$En[K_S , M] \parallel En[K_B , (K_S \parallel ID_A)]$
Request Random Session Key
$ID_A \parallel ID_B$
KDC Server
$En[K_A , (T_S\| K_S)] \parallel$
$En[K_B , (K_S \parallel ID_A)]$
Generate Random Session Key K_S

Figure 2. Send E-mail Process Diagram

2-2. RECEIVE E-MAIL PROCEESS

To decrypt the e-mail by CA method, the receiver inserts the smart card into the reader and enters his PIN. The encrypted K_S is decrypted with receiver's private key, which is stored in his smart card. Then, the K_S is used to decrypt the e-mail. To decrypt the digital signature, sender's public key is retrieved from the corporate directory where her certificate is published. Sender's public key may also be sent with the e-mail message, depending on the security configuration of the client. The same hash algorithm generated when sender inserted the smart card into the reader is used to produce a message digest locally. If the received message digest and the locally generated message digest match, the mail has not been altered during transport and sender is authenticated as the author of the e-mail message.

$$K_S=P_{priB} [P_{pubB}[K_S]]$$

```
$M=De[K_S , M]$
IF $P_{pubA} [H(M)] = H(M)$ THEN
    Message matched
ELSE
    Message had be tampered
```

The e-mail that is received with the method of KDC, the system decrypted the session key K_S and sender's ID_A by master key K_B of receiver in the smart card, and confirms the sender's ID_A in order to achieve no denying. Then the system decrypts the message by K_S.

$$K_S \| ID_A = De[K_B, (K_S \| ID_A)]$$
$$M = De[K_S , M]$$

The receiving e-mail process diagram is illustrated in figure 3.

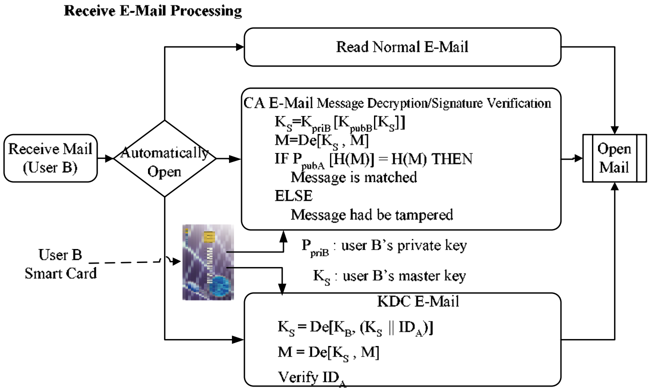

Receive E-Mail Processing
Read Normal E-Mail
CA E-Mail Message Decryption/Signature Verification
$K_S=K_{priB} [K_{pubB}[K_S]]$
$M=De[K_S , M]$
IF $P_{pubA} [H(M)] = H(M)$ THEN
Message is matched
ELSE
Message had be tampered
Receive Mail (User B)
Automatically Open
Open Mail
User B Smart Card
P_{priB} : user B's private key
K_S : user B's master key
KDC E-Mail
$K_S = De[K_B, (K_S \| ID_A)]$
$M = De[K_S , M]$
Verify ID_A

Figure 3 Receive E-mail Process Diagram

III. DEVELOPEMENT OF THE SMART CARDS

The life of storage devices, such as hard disks, floppies, is only for one to three years. They are easy to dampened or lost. The smart card can preserve the data over five years, and it can be protected [8]. The private key of certificate and master key of KDC's user store in EEPROM that has own file system. The smart card provides the RSA algorithm. When CA enrolls the certificates of users, the public-key is generate on the smart card to prevent it to be tamper. The smart card has two-factor authentication, one is base on card's PIN and the other is card itself. The users could enter the PIN for three times and the card will be locked or failed. The users cannot also copy the keys of the smart cards. The hardware or operating system of the smart card prevents to be attack. A key benefit of smart cards is in public key cryptography where the private key and public key certificate of the user can be stored on the smart card. Operations that require the use of the private key can performed on the smart card itself thus isolating any security-critical computations. The credentials on the smart card are portable so for example, an employee would be able to sign their e-mail at any company workstation that has a smart card reader installed and not be limited to their own. In addition, smart cards are tamper resistant with built-in physical defenses.

In order to enable smart card's function of software data storage, the computer software must contain standardized Cryptographic Application Interface (CAPI). CAPI can avoid the inconsistency of different public-key systems developed by various system developers. By establishing a standard model for each mandatory function, CAPI provides application developers to reduce the development cost about public-key systems. However, even as the platform had standardized, the hardware tokens provided by each vendor may vary. As a result, each vendor must supply the corresponding platform to each hardware token, call Cryptographic Sevices Provider (CSP).

The most common CAPI's on the market today are the MS CAPI and the PKCS #11 of RSA. The Smartcard-based Secure E-mail System recommended in our development adopts the smart card Chip Operation System (COS), developed by NexSmart Technology Incorporation, as the hardware token. This specific COS, named NexCard 2.0, used the Windows for Smart Card (WfSC) chip operation system of MS Windows as its core structure [10]. The NexCard has 8051-compatible microprocessor, 136Kbytes mask-ROM, 32Kbytes EEPROM, 5Kbytes RAM and I/O port. The mask-ROM contains the COS, and it is etched during manufacture. The CSP provided by MS for WFSC 1.1 does not have the function of producing RSA key pairs on card. Besides, Key pairs produced by external programs of COS cannot meet the safety standard while encoding onto the off card key generation. Therefore, we also developed CSP 2.0 that is capable of on card key generation for WfSC CAPI.

IV. SYSTEM IMPLEMENTATIONS

For the implementations of Smartcard-based Secure E-mail System, the direction can be given according to the independent operation of CA and KDC as below.

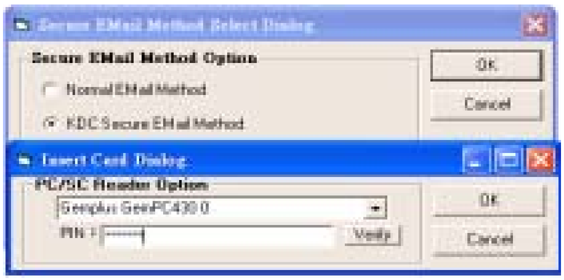
4-1. IMPLEMENTATIONS OF CA

The MS CA simulates trusted third-part CA [10]. It uses the CAPI of NexCard, and through the RSA co-processor on card it produces 512 or 1024 bits asymmetric key pairs on smart card. The private key will be stored directly on smart cards, while the publish keys and certificate will be stored on repositories of certificates for others' signatures. In MS Outlook, the default certificate will be used automatically while secure mails are being sent out. If this certificate is to be stored on smart card, Outlook will request an insert of smart card. User can then enter PIN code in order to access the private key on smart key to sign confidential signature. The certificate of the receiver will be certainly stored in contact's mailbox. However, Outlook will not check the validity of such certificate automatically. Therefore, user must download and enter invalidation certificates manually and frequently. To overcome such problem, we had added property page in MS Outlook to enable user to pre-set favourite CA CRL addresses. Whenever MS Outlook is activated, it will download and enter CRL automatically to confirm the validity of certificates. When an e-mail is sent out by MS Outlook, it will be transformed to the needed format for S/MIME automatically. As long as the e-mail system of the receiver can support S/MIME, e-mails can be displayed with no difficulties.

4-2. IMPLEMENTATIONS OF KDC

KDC is mainly responsible of card issuance, card management and random session key generation. As a result, the safety of KDC itself is very important. When KDC is doing the issuance or operations, it must use smart card to log on. As the card is removed, it will also log out the system automatically. In terms of delivering random session key by KDC, it needs to use multi-threads to respond to the request of multiple users simultaneously. The data about session key is sent by encrypted network packets. Moreover, it can set TCP port to increase the safety of delivery.

We implement the plug-in library in MS Outlook. This library includes a few parts. One of the parts uses CAPI from NexCard to do precise calculations such as DES, triple-DES, AES. In addition, it uses smart card hardware such as PC/SC reader. Second part is the KDC secure e-mail processing flow. We will process the previously defined procedures in the component. Defined procedures include the received session key and time stamp between KDC, mail header will define Content-Type to differentiate KDC mails, and to secure the content by Advanced Encrypting Standard (AES) algorithm [11]. Just call the API functions, secured logic calculation and usage of smart card both are provide in the Cryptographic Application Interface. When mail opened at the receiver's end, the mail will automatically define the transmission method, and get the random session key to decrypt the mail addition, confirm sender's ID is the same as the decrypted ID to see if the mail is sends by hackers. The third part is the user interface design. This mainly provides choices for users to choose transmission methods such as normal, CA, or KDC methods when sending an-e-mail. For example, KDC method will require KDC to call for KDC e-mail processing component to get the random session key process related procedures. In addition, this will include property page setup and TCP port. CA's CRL location between KDC and secured encryption calculation related setups. We pick the image from the KDC sending e-mail methods such as figure 4.



Secure EMail Method Select Dialog

Secure EMail Method Option

○ Normal EMail Method

● KDC Secure EMail Method

Insert Card Dialog

PC/SC Reader Option

Gemplus GemPC430 0

PIN : ••••••

Figure 4. KDC send e-mail method image

V. CONCLUSIONS

Although under the mechanism of PKI, CA certified safely transmitted mail is widely being used. To use smart card to save private key's importance is nowadays promote. But in the corporation where not too much man power to manage the complex CA mechanism, choose to use KDC procedure can efficiently process secure e-mail. This is a good choice because it combines CA certificate and KDC's master key onto one smart card. We will continue to develop the usage of smart card in the web mail to get KDC security, to use the Pocket PC combined smart card reader and wireless network modules (GPRS or wireless network interface) to send secure e-mail for mobile users, and to use one master key to secure files. This will allow users not to memorize different passwords, and just to remember the PIN code on the smart card. We can also smart card as a USB token, and include related applications in the USB flash memory to allow mobile users to send out secure e-mails at any time without worrying the security of the e-mail contents.

VI. REFERENCES

[1]Stephen T. Kent, "Internet Privacy Enhanced Mail," *Communications of the ACM*, Vol. 36, No. 8, August 1993, pp. 48-60.

[2]Bruce Schneier, *E-Mail Security: How to Keep Your Electronic Messages Private*, John Wiley & Sons, Inc., 1995.

[3]Philip Zimmermann, "T*he Official PGP User's Guide"*, MIT Press, 1995.

[4]John Linn, "Privacy Enhancement for Internet Electronic Mail, Part I: Message Encryption and Authentication Procedures," RFC 1421, Feb. 1993.

[5]Blake Ramsdell, "S/MIME Version 3 Message Specification," RFC 2633, June 1999.

[6]William Stallings, *Cryptography and Network Security: Principles and Practice*, 3rd edition, Prentice-Hall, Inc. 2002.

[7]Anusha Nirmalananthan, "The Smart Card Cryptographic Service Provider Cookbook", Microsoft Corporation, October 2002.

[8]Wolfgang Rankl and Wolfgang Effing, *Smart Card Handbook*, 2nd edition, John Wiley & Sons, 2000.

[9]Nexsmart NexCard 2.0, NexSmart Technology, Inc., http://www.nexsmart.com/nexcard

[10]Microsoft Corporation, The Smart Card Deployment Cookbook, http://www.microsoft.com/technet/security/prodtech/smrtcard/smrtcdcb/default.asp

[11]National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," Federal Information Processing Standard, *FIPS PUB 197*, November 26, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf